

Visual Modeler



Guide des bonnes pratiques

Release 9.1

Visual Modeler



Guide des bonnes pratiques

Release 9.1

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant à la section «Remarques», à la page 43.

Copyright

La présente édition s'applique à la version 9.1 de Visual Modeler et à toutes les éditions et modifications ultérieures, sauf mention contraire dans les nouvelles éditions.

© Copyright IBM Corporation 2007, 2011.

Table des matières

Chapitre 1. Bonnes pratiques en matière de sauvegarde et de restauration 1

Présentation de l'architecture de déploiement	1
Infrastructure de déploiement	2
Stratégies de sauvegarde	3
Stratégie de restauration de la base de données. . . .	3
Récupération de la base de données	4
Restauration du serveur d'applications et du serveur Web	4

Chapitre 2. Comprendre les icônes de l'interface utilisateur 5

Chapitre 3. Modélisation des éléments . 7

Utilisation des chemins relatifs et absolus pour la modélisation d'un élément.	7
Planification de la modélisation des éléments	7
Considérations liées à la conception pour la modélisation des éléments.	8
Définition de la taille d'un modèle d'élément	8
Utilisation des commandes contextuelles pour spécifier la quantité d'éléments	10
Définition des propriétés d'un modèle	12
Définition des noms des propriétés d'un modèle	12
Définition de propriétés de modèle uniques	13
Définition de la hiérarchie des propriétés d'un modèle	14
Définition de plusieurs propriétés avec la même valeur	14
Utilisation des feuilles de travail pour l'affectation des propriétés	15
Éviter le chaînage dans les formules de propriétés	16
Définition des règles	16
Définition des conditions de règle	16
Classement des fragments de règle	17
Création de règles générales.	17
Utilisation des formules pour définir des règles	17
Éviter les informations de chemin dans les règles	18
Comparaison entre les tableaux de contraintes et les règles	18
Amélioration des performances des modèles	19

Outils de modélisation.	19
Utilisation du journal de trace	20
Utilisation de l'outil de génération de rapports sur les modèles	22
Utilisation des outils de test de charge	22
Outil d'affichage de l'état du cache	24
Performances.	24
Règles de performances d'un modèle.	24
Propriétés de performances des modèles.	25

Chapitre 4. Archivage des données . . . 27

Chapitre 5. Mise à jour des statistiques 29

Mise à jour des statistiques pour une base de données Oracle	29
Mise à jour des statistiques pour une base de données SQL Server	29

Chapitre 6. Instructions concernant le réglage et la mémoire de la machine virtuelle Java 31

Ajustement des paramètres de mémoire de la machine virtuelle Java	31
Réglage complémentaire des performances	32
Traçage des activités de récupération de place. . . .	32

Chapitre 7. Analyseur de journal. . . . 35

Configuration des rapports quotidiens de l'analyseur de journal	36
Automatisation des rapports quotidiens de l'analyseur de journal	36
Configuration des rapports quotidiens de l'analyseur de journal	37
Recommandations concernant la structure du répertoire	37
Configuration de l'analyseur de journal	38

Index 41

Remarques 43

Chapitre 1. Bonnes pratiques en matière de sauvegarde et de restauration

Les meilleurs plans de restauration mettent l'accent sur la prévention. La configuration d'un environnement solide, la mise en place de systèmes redondants, l'élaboration régulière de politiques de sauvegarde et de restauration et la réalisation de tests réguliers pour vous assurer que vous pouvez restaurer des données à partir de la sauvegarde permettent de limiter les conséquences d'un sinistre en protégeant chaque couche de votre infrastructure de déploiement.

Certaines décisions concernant les politiques de sauvegarde et de restauration doivent être basées sur des critères commerciaux. Quelle est la valeur du site disponible par jour pour votre activité ? Quelles quantités de données pouvez-vous vous permettre de perdre ? Pendant combien de temps votre site de e-commerce peut-il rester indisponible pour les clients ? Quels sont les avantages et les inconvénients en termes de temps et de coût pour les différentes solutions de sauvegarde et de restauration ? Les réponses à ces questions vous aideront à déterminer vos exigences en matière de sauvegarde et de restauration.

Le système de sauvegarde le plus simple peut consister à sauvegarder les données et une copie de votre application sur une bande magnétique ou sur d'autres unités distantes et à stocker les données dans un centre de données à distance. Une meilleure stratégie est d'installer des systèmes redondants pour chaque partie de votre déploiement de sorte à s'assurer que si l'un des systèmes est défaillant, un autre est déjà disponible. La solution la plus solide est de conserver une image miroir de votre site à un emplacement distant et de synchroniser régulièrement cette image avec les données actives. Cette dernière solution est la plus coûteuse mais elle permet une restauration immédiate. Les solutions précédentes sont moins chères mais nécessitent plus de temps et d'effort pour effectuer la restauration.

Présentation de l'architecture de déploiement

Le fait de configurer un environnement de déploiement riche permet non seulement de faciliter les mises à jour et la maintenance du site, mais cela fournit également un moyen de reconstruire rapidement une application complète dans le cadre d'une étape de restauration en cas de besoin. L'architecture de déploiement est composée des éléments suivants :

- Environnement de construction : Un environnement de construction connu et prévisible contenant tous les éléments nécessaires à la construction du déploiement, à savoir :
 - les kits Java Development Kit
 - les kits SDK
 - le référentiel de codes (comme par exemple le système CVS)
 - les bibliothèquesLes étapes nécessaires pour passer du code à l'étape de production sont variables et peuvent être itératives ; pour autant, l'environnement de construction doit contenir tous les éléments requis pour reconstruire votre déploiement de manière prévisible en cas de besoin.
- Zone d'assurance qualité : Un environnement distinct utilisé pour effectuer des tâches en lien avec l'assurance qualité. La zone d'assurance qualité est le premier

environnement dans lequel le travail provenant (éventuellement) de plusieurs ingénieurs est intégré pour fonctionner comme une seule unité.

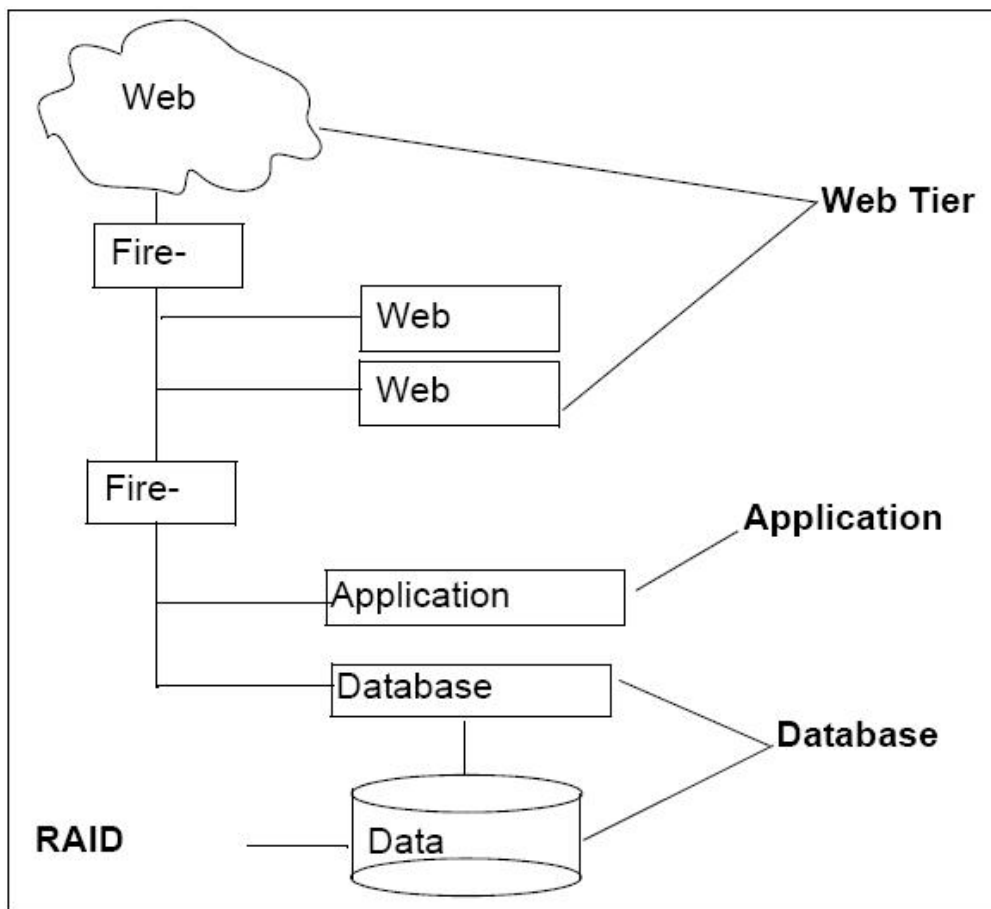
- Zone de transfert : Un environnement distinct utilisé pour exécuter le travail intégré dans la zone d'assurance qualité dans un contexte spécifique qui émule la production.

Infrastructure de déploiement

Une stratégie courante utilisée pour établir une infrastructure solide est de "tout doubler", c'est-à-dire mettre en place des systèmes redondants de sorte à s'assurer que lorsqu'un système principal est défaillant, un système secondaire est immédiatement disponible en ligne. La figure suivante présente une infrastructure typique composée de trois niveaux :

- Niveau Web : Tous les composants gérant les demandes provenant de navigateurs Web et fournissant des contenus à des navigateurs Web.
- Niveau application : Tous les composants traitant les demandes provenant du niveau Web et fournissant des contenus dynamiques au niveau Web, en utilisant généralement des données provenant du niveau base de données.
- Niveau base de données : Tous les composants fournissant des données au niveau application.

Le diagramme suivant présente une infrastructure de déploiement typique :



La présence de deux serveurs Web garantit que si l'un est défaillant, l'autre peut continuer de fonctionner. La mise en place d'un deuxième pare-feu pour fournir

une protection supplémentaire des données satisfait également à certaines exigences réglementaires en matière de sécurité des données.

Pour protéger physiquement vos données, une des stratégies consiste à stocker les données de production sur un périphérique RAID. Si une seule unité est défaillante, aucune donnée ne sera perdue. Il existe un seuil mécanique lorsque vous utilisez ce type de stratégie : selon votre configuration, si le nombre d'unités défaillantes est supérieur à ce seuil, des données seront perdues. Vous devez tenir compte de cet aspect lorsque vous déterminez vos exigences.

Stratégies de sauvegarde

Il existe différentes stratégies de sauvegarde pour protéger les données et récupérer vos serveurs d'applications et serveurs Web. La plupart des stratégies de sauvegarde consistent à enregistrer des copies des opérations exécutées au niveau application et au niveau Web.

Stratégie de restauration de la base de données

La stratégie de sauvegarde détermine la rapidité avec laquelle vous pouvez restaurer les données après un sinistre. Déterminez un diagramme approprié pour rendre votre base de données à nouveau opérationnelle, en incluant le temps nécessaire pour reconstruire le système d'exploitation et recharger la base de données si besoin, et prévoyez votre politique de sauvegarde en conséquence.

Vous trouverez ci-après les types de sauvegardes de la base de données que vous devez effectuer :

- Sauvegardes de point de contrôle : Les serveurs de base de données consignent l'activité survenue au niveau des transactions au fur et à mesure que les transactions sont validées. Les sauvegardes de point de contrôle écrivent un journal des transactions à partir de la dernière fois qu'une sauvegarde de point de contrôle a été réalisée. Écrivez le journal de point de contrôle sur une unité physique distincte. Cela permet de créer une image instantanée de l'activité de la base de données au niveau des transactions. En cas d'échec de la base de données, il existe une chaîne d'enregistrements qui permettent de reconstruire l'activité.
- La fréquence à laquelle votre déploiement effectue les sauvegardes de point de contrôle est une décision de votre entreprise. Si les transactions réalisées sur votre site s'élèvent à plusieurs millions de dollars par heure, il est conseillé d'effectuer des sauvegardes de point de contrôle plusieurs fois par heure. En revanche, si le nombre de transactions par heure effectuées sur votre site est bas, les sauvegardes de point de contrôle peuvent être moins fréquentes. Déterminez un intervalle qui permet de restaurer les données en fonction de votre niveau d'activité commerciale.
- Sauvegardes incrémentielles quotidiennes : Les sauvegardes incrémentielles enregistrent uniquement les fichiers qui sont modifiés tous les jours. Réaliser des sauvegardes incrémentielles quotidiennes sur une machine différente, plutôt qu'utiliser un support physique, est une bonne stratégie. La sauvegarde est en fait une copie disque à disque.
- Sauvegardes hebdomadaires complètes : Les sauvegardes complètes enregistrent l'intégralité de la base de données, pas seulement les fichiers qui ont été modifié depuis la dernière sauvegarde. Réaliser des sauvegardes complètes sur une machine différente est une bonne stratégie.

Récupération de la base de données

Pourquoi et quand exécuter cette tâche

Une récupération typique de la base de données implique les étapes suivantes :

Procédure

1. Restaurer la base de données initiale à partir de la dernière sauvegarde des données.
2. Restaurer les données par ordre de date à partir des sauvegardes incrémentielles quotidiennes.
3. Reconstruire l'activité des dernières heures à l'aide des sauvegardes de point de contrôle.

Restauration du serveur d'applications et du serveur Web

Lorsque vous élaborez des politiques de restauration pour votre serveur d'applications et votre serveur Web, prévoyez de construire des remplacements exacts qui pourront être utilisés en cas de défaillance du serveur d'applications ou du serveur Web, conformément au principe de la reconstruction et de la remise en état de fonctionnement immédiates.


Assurez-vous d'avoir des copies de tout ce qui rend les déploiements de votre serveur d'applications et de votre serveur Web uniques : fichiers de configuration, fichiers de propriétés, machine virtuelle Java, code source original de votre référentiel CVS, etc. Sauvegardez toutes les données statiques conservées sur le niveau Web, comme par exemple les pages JSP personnalisées. Assurez-vous que votre processus de sauvegarde inclut les fichiers de configuration du conteneur ou du serveur Web, ou d'autres fichiers nécessaires au fonctionnement de votre site, mais qui ne peuvent pas être considérés comme code source et ne sont donc pas stockés dans un référentiel de code source.

Utilisez vos environnements d'assurance qualité et de transfert comme points de départ pour reconstruire vos serveurs et revenir à un état opérationnel.

Chapitre 2. Comprendre les icônes de l'interface utilisateur

Le tableau suivant décrit les icônes utilisées dans Visual Modeler

Tableau 1. Tableau des icônes

Icône	Description
	Modifier le modèle - Cette icône est utilisée pour modifier un modèle.

Chapitre 3. Modélisation des éléments

Utilisation des chemins relatifs et absolus pour la modélisation d'un élément

Cette rubrique présente l'utilisation des chemins relatifs ou absolus pour spécifier des entités telles que des propriétés et des règles. Les chemins ont la forme suivante :

<noeud du groupe de modèles racine>.<chemin de l'élément d'option auquel s'applique la propriété ou la règle>.<nom de la propriété ou de la règle>

Par exemple, prenons le chemin absolu suivant vers la propriété `memoryProvided` :

`MXDS-7500.memory.sim256.memoryProvided`

Le noeud racine du groupe de modèles est `MXDS-7500` ; le chemin vers l'élément option auquel s'applique la propriété `memoryProvided` est `memory.sim256` ; `memoryProvided` est le nom de la propriété.

Si vous prévoyez d'utiliser une propriété ou une règle dans plusieurs modèles, vous pouvez utiliser des symboles spéciaux pour spécifier les chemins relatifs. Par exemple, le symbole "*" dans le chemin suivant indique que le chemin commence à la racine de la hiérarchie des groupes de modèles :

`*.memory.sim256.memoryProvided`

L'utilisation d'un point (.) indique que le chemin commence à partir du point d'attache de la règle. Par exemple, le "." dans le chemin suivant indique qu'il y a un élément d'option appelé `sim256` dans une classe d'option appelée "memory" dans le modèle en cours.

`.memory.sim256`

Planification de la modélisation des éléments

Un modèle représente un élément configurable. Lorsque vous vous préparez à planifier une implémentation d'IBM Sterling Web, vous commencez par réfléchir à la conception d'un modèle de l'élément. Il n'existe pas de "bonne" manière pour modéliser un élément particulier. D'un autre côté, il existe une multitude de manières de créer des modèles qui, bien que correctes d'un point de vue technique, sont inefficaces et difficiles à maintenir.

Vous trouverez ci-après une liste des avantages et des inconvénients que vous devez garder à l'esprit.

- Facteurs liés au coût : Création, maintenance et performances.

Vous devez comparer le coût de création du modèle avec le coût de maintenance et les performances attendues. Le coût de création du modèle représente les efforts dépensés de manière ponctuelle pour le développer, le coût de maintenance du modèle représente les efforts dépensés au fil du temps pour la maintenir et l'améliorer, alors que les performances représentent la vitesse d'exécution du modèle sur une plateforme matérielle spécifique. Vous pouvez

optimiser l'un de ces facteurs, mais n'oubliez pas que si vous essayez d'en optimiser plus d'un, cela signifie en général vous avez des objectifs contradictoires. Par exemple, un modèle complexe peut s'exécuter rapidement, mais être difficile à maintenir. Pour une présentation des modèles et des bonnes pratiques en matière conception, voir «Considérations liées à la conception pour la modélisation des éléments».

- Rôles de l'implémenteur : Générateur de modèles uniquement, générateur et chargé de la maintenance des modèles, chargé de la maintenance des modèles uniquement.

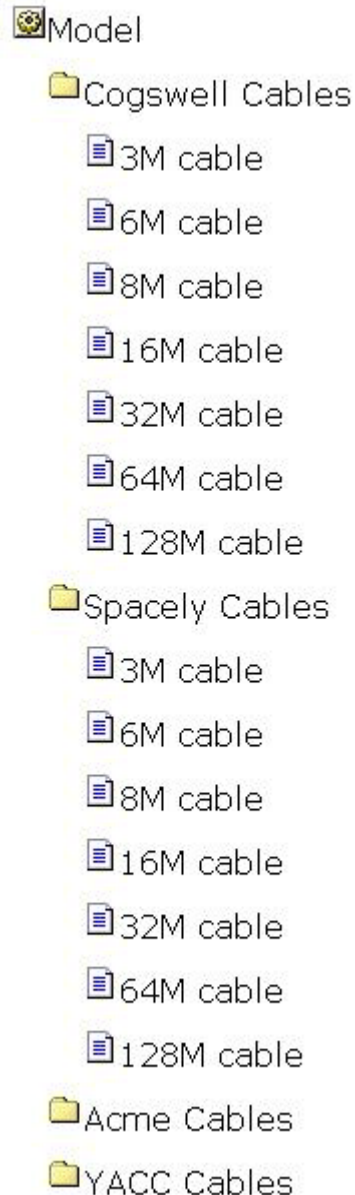
La conception d'un modèle est influencée par le rôle de l'implémenteur. Par exemple, un consultant à qui l'on donne un mois pour implémenter un modèle qui sera ensuite remis à un autre groupe pour la maintenance peut mettre l'accent sur la conception rapide du modèle, plutôt que sur la conception d'un modèle qui sera facile à maintenir. En revanche, si l'implémenteur est également chargé de la maintenance, il faudra peut-être plus de temps pour concevoir le modèle et ce dernier ne s'exécutera peut-être pas aussi rapidement, mais il sera plus facile à maintenir sur le long terme. Quel que soit votre rôle, votre objectif doit être de configurer un modèle qui ne posera pas de problèmes par la suite.

Considérations liées à la conception pour la modélisation des éléments

Lors de la conception d'un modèle d'élément, il est important de mettre en pratique des principes de conception qui permettent à un modèle complexe de fonctionner efficacement.

Définition de la taille d'un modèle d'élément

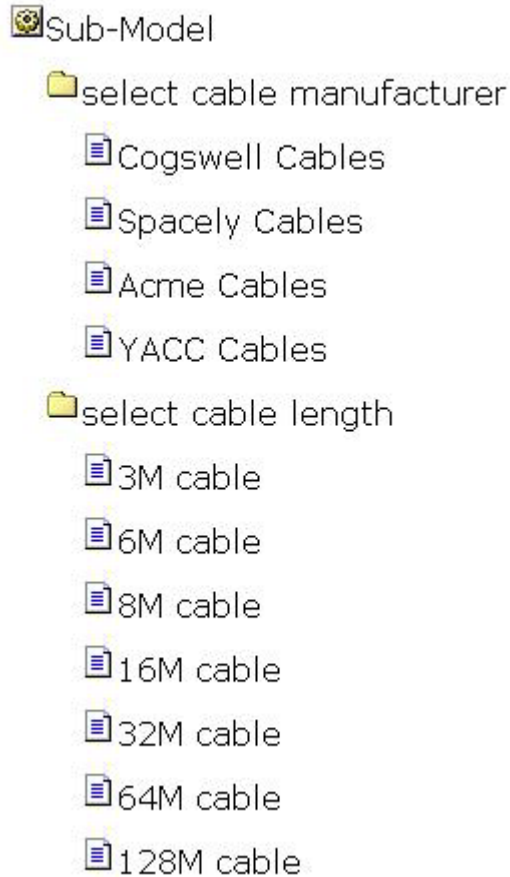
La taille d'un modèle revêt une importance particulière. Les modèles de grande taille sont plus longs à restituer dans le navigateur, ils sont difficiles à maintenir et pendant la configuration, le configurateur parcourt la structure du modèle plusieurs fois pour obtenir les prix, les règles de déclenchement, etc. Vous pouvez améliorer les performances et diminuer les coûts de maintenance en gardant la taille d'un modèle aussi petite que possible via l'utilisation de sous-assemblages et autres techniques décrites dans la présente rubrique. Par exemple, supposez que vous ayez plusieurs fournisseurs de câbles, chacun d'eux proposant différentes longueurs de câbles. Vous souhaitez donner à l'utilisateur la possibilité de sélectionner la quantité, la longueur et le fournisseur des câbles. Pour ce faire, vous pouvez créer des éléments d'option dans votre modèle pour représenter chacune des options disponibles, comme indiqué dans la figure suivante :



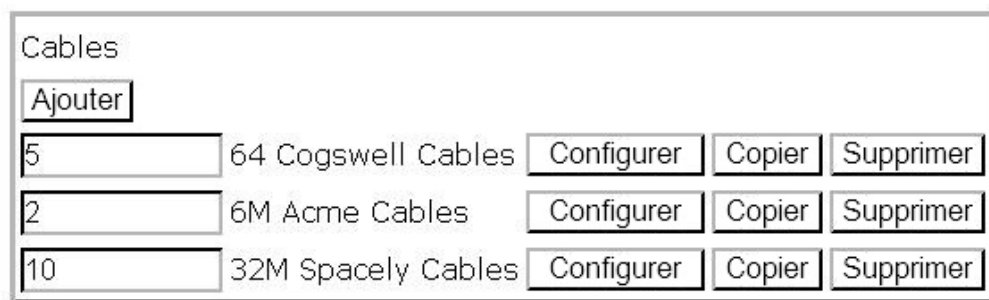
Cette approche va fonctionner, mais il peut être fastidieux d'implémenter, maintenir et créer un modèle avec de nombreuses options qui ne présente pas d'intérêt pour l'utilisateur final.

Une autre approche serait d'implémenter les différents éléments d'option de longueurs de câbles en tant que groupe d'éléments d'option, puis d'inclure ce groupe sous chacun des fabricants. Cela facilite la maintenance, car le modélisateur peut mettre à jour les informations sur les éléments d'option des câbles à partir d'un seul emplacement. Toutefois, si vous optez pour cette approche, la liste dans laquelle l'utilisateur final doit choisir est énorme et les performances ne sont pas améliorées puisque le modèle à parcourir est toujours volumineux.

Une meilleure option est de créer un sous-modèle permettant à l'utilisateur de sélectionner un fabricant de câbles et la longueur souhaitée, puis d'utiliser l'instanciation dynamique pour permettre à l'utilisateur d'ajouter autant de types et de longueurs de câbles que nécessaire, comme indiqué dans la figure suivante :



La figure suivante présente un exemple d'interface utilisateur de sélection de câbles qui utilise une instanciation dynamique pour permettre aux utilisateurs finals de configurer leur sélection de câbles.



Comme vous pouvez le constater, cette approche permet de conserver un modèle de petite taille. La maintenance de ce modèle sera plus facile car le modélisateur n'a plus à traiter un grand nombre d'éléments d'option dupliqués, les performances sont améliorées car le modèle est plus petit et la configuration est plus facile pour l'utilisateur final puisqu'il n'a plus à choisir parmi une longue liste de types de câbles et de fabricants.

Utilisation des commandes contextuelles pour spécifier la quantité d'éléments

Il arrive parfois que le modélisateur souhaite permettre aux utilisateurs de sélectionner un élément, puis d'entrer le nombre d'éléments souhaité. Dans ce cas, la meilleure façon de procéder est de définir l'affichage de classe d'option sur

popup-qty. Lorsque l'utilisateur final sélectionne un élément, une zone de quantité apparaît, permettant à l'utilisateur final d'entrer le nombre d'éléments souhaités.

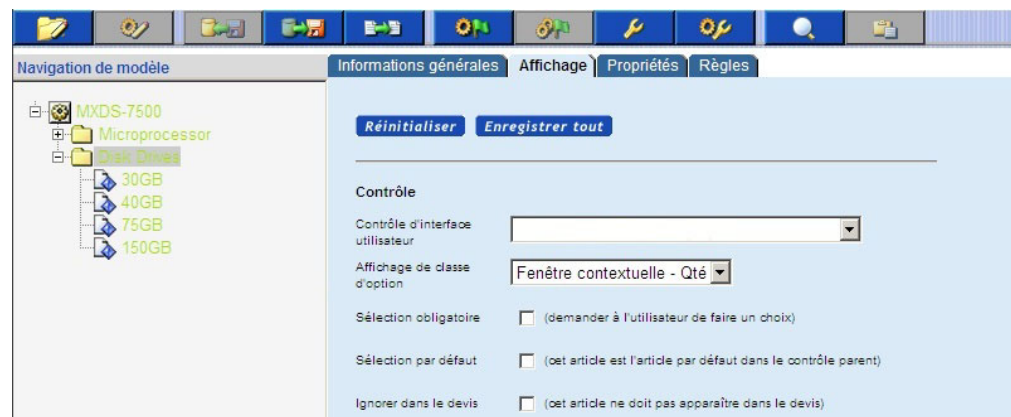
À la place de la zone de quantité, certains modélisateurs préfèrent utiliser une commande qui récupère les valeurs entrées par l'utilisateur, en affichant une zone de texte à côté de l'élément, dans laquelle les utilisateurs peuvent saisir une quantité. Le problème est que le comportement de la commande popup-qty diffère considérablement du comportement d'une commande qui récupère les valeurs entrées par l'utilisateur : si la première comporte un traitement de la quantité intégré, la deuxième requiert des manipulations supplémentaires.

Lorsqu'un utilisateur final entre une quantité dans une zone popup-qty, l'application sélectionne automatiquement la quantité de l'élément sélectionné. Toutes les propriétés associées à cet élément sont comprises dans l'état du configurateur (pool de propriétés) et les valeurs des propriétés numériques sont multipliées par la quantité entrée.

Lorsqu'une valeur est entrée dans une commande de type zone de texte, aucun traitement n'est effectué. Les commandes récupérant les valeurs entrées par l'utilisateur ont été conçues simplement pour stocker certaines informations complémentaires sur l'utilisateur. Pour faire en sorte que cette commande se comporte comme une zone de quantité, le modélisateur doit écrire une règle d'extension qui récupère la valeur entrée par l'utilisateur et sélectionne un nombre d'éléments égal à cette valeur. Vous ne pouvez pas utiliser la valeur entrée par l'utilisateur dans la zone de texte pour définir la propriété `_quantity` à l'aide d'une règle d'affectation, puisque cela ne crée pas automatiquement des instances des propriétés de l'élément dans le pool de propriétés.

Pour afficher une zone popup-qty à côté de l'élément sélectionné, utilisez le style d'affichage de classe d'option popup-qty et l'un des affichages tabulaires avec des commandes de quantité. Cela permet de s'assurer que le nombre approprié d'éléments est sélectionné et que les propriétés appropriées sont copiées dans le pool de propriétés.

Par exemple, la figure suivante montre comment configurer une commande popup-qty à l'aide de Visual Modeler. À partir du panneau Modèles et groupes, sélectionnez le modèle que vous souhaitez modifier, puis cliquez sur l'icône **Modifier le modèle**. La page Navigation de modèle s'affiche. Cliquez sur le groupe d'options que vous souhaitez modifier, cliquez sur l'onglet **Affichage** puis sélectionnez l'affichage tabulaire de sélection multiple dans la liste déroulante **Contrôle d'interface utilisateur**, comme indiqué sur la figure suivante :



Faites défiler jusqu'en bas de la page et entrez les paramètres En-têtes de colonne, Propriétés des colonnes et Alignement des colonnes. La figure suivante présente des exemples de paramètres :



Pour terminer, compilez et testez le modèle. La commande popup-qty doit être placée à l'endroit spécifié sur la page Product Configurator.










Définition des propriétés d'un modèle

L'utilisation des propriétés est très courante dans IBM Sterling Configurator. Les modélisateurs associent des propriétés aux modèles, classes d'option et éléments d'option, puis ils écrivent des règles applicables à ces propriétés pour afficher des messages, afficher, masquer ou sélectionner des éléments, voire même définir les valeurs d'autres propriétés. Étant donné que les propriétés jouent un rôle considérable dans la modélisation d'un élément configurable, il est important qu'elles soient définies et utilisées avec soin. Cette section présente des astuces utiles et des procédures à suivre lorsque vous définissez et associez des propriétés.








Définition des noms des propriétés d'un modèle

Parfois lors du développement d'un modèle, surtout lorsque les délais sont serrés, le modélisateur est tenté de prendre des raccourcis pour accélérer le processus de développement. Un des raccourcis les plus courants est de créer des propriétés avec des noms courts et souvent vagues ou peu explicites. Bien que cela permette d'accélérer le développement du modèle sur le court terme, cette pratique augmente considérablement les efforts requis pour la maintenance du modèle. Lors de la conception des modèles, le modélisateur doit toujours attribuer des noms de propriétés permettant de comprendre immédiatement la signification d'une propriété donnée. Plus le nom que vous donnez à une propriété est explicite, plus il est facile de déboguer et maintenir le modèle au moment de la conception et par la suite.

Prenons l'exemple suivant :

model node	properties
 Model	$tmr = \text{sum}(mr)$ $tmp = \text{sum}(mp)$ $amr = \text{value}(tmr) - \text{value}(tmp)$
 OC1	 oi1 mr=1  oi2 mr=2
 OC2	 oi3 mp=2  oi4 mp=1

Au premier coup d'oeil, il n'est pas évident de savoir à quoi correspondent les propriétés affectées à ce modèle. En prenant le temps de créer des noms plus significatifs, il devient bien plus facile de comprendre rapidement les fonctions de toutes les propriétés et les rapports qui les relient entre elles.

model node	properties
 Model	$\text{total_mem_required} = \text{sum}(\text{mem_required})$ $\text{total_mem_provided} = \text{sum}(\text{mem_provided})$ $\text{additional_mem_required} = \text{value}(\text{total_mem_required}) - \text{value}(\text{total_mem_provided})$
 OC1	 oi1 mem_required=1  oi2 mem_required=2
 OC2	 oi3 mem_provided=2  oi4 mem_provided=1

Définition de propriétés de modèle uniques

Afin d'implémenter rapidement une fonction, il arrive souvent qu'un modélisateur réutilise une propriété existante au lieu de créer une nouvelle propriété conçue spécifiquement pour le problème à traiter. Cela peut entraîner les deux conséquences suivantes :

- Le modèle peut être plus difficile à comprendre si le nom de la propriété existante n'a pas de lien avec le problème à traiter.
- La réutilisation d'un nom de propriété peut entraîner des erreurs dans le modèle si elle entre en conflit avec l'utilisation d'origine de la propriété.

Revenons sur l'exemple utilisé dans la section précédente. Supposez que le modélisateur a créé une propriété pour stocker la mémoire requise, qu'il a appelée "mémoire" (voir la section précédente pour comprendre pourquoi ce premier choix est inapproprié). Lorsqu'il a besoin d'une nouvelle propriété pour stocker la mémoire fournie, il se rend compte qu'il a déjà créé une propriété appelée "mémoire" et décide d'utiliser celle-ci au lieu de créer une nouvelle propriété.

Au premier coup d'oeil, il semble que tous les éléments d'option requièrent une quantité de mémoire, et non que deux éléments requièrent de la mémoire et que deux éléments fournissent de la mémoire. En plus de cela, la propriété

mémoire_totale_requise n'aura plus la valeur appropriée puisqu'elle correspond maintenant à la somme de deux mémoires - mémoire requise et mémoire fournie. Si la modélisation a été effectuée de cette manière, le modélisateur doit effectuer un travail supplémentaire pour séparer les instances spécifiques des propriétés dont il a besoin, en utilisant les chemins d'accès complets ou relatifs des éléments contenant les instances de propriété appropriées.

Définition de la hiérarchie des propriétés d'un modèle

Les propriétés peuvent être définies à tous les niveaux de la hiérarchie du groupe de modèles, depuis le niveau du groupe de modèles racine au niveau d'un modèle individuel. Le niveau auquel une propriété est définie détermine quels modèles peuvent voir et utiliser cette propriété. En concevant attentivement vos modèles, vous pouvez accélérer le développement des modèles et simplifier l'organisation des propriétés. Utilisez les instructions suivantes pour déterminer à quel niveau une propriété doit être définie :

Si une propriété est destinée à être seulement utilisée dans un modèle particulier, définissez cette propriété au niveau du modèle.








Si une propriété est destinée à être utilisée dans plusieurs modèles au sein d'un groupe particulier de modèles, définissez cette propriété au niveau du groupe de modèles.

Si une propriété est destinée à être utilisée dans des modèles qui étendent des groupes de modèles, définissez cette propriété au niveau du premier groupe de modèles qui contient tous les groupes de modèles dont les modèles utiliseront la propriété.

En dernier lieu, définissez la propriété au niveau du noeud du groupe de modèles racine.

Définition de plusieurs propriétés avec la même valeur

Lorsque plusieurs propriétés ont la même valeur, le modèle peut être plus facile à construire et maintenir. À premier abord, ce concept peut être source de confusion et sera mieux illustré avec un exemple. Supposez que vous construisez un modèle qui permette à l'utilisateur de faire un choix parmi une sélection de batteries de disques. Un certain nombre de disques sont associés à chaque type de batterie de disques. L'utilisateur peut choisir plusieurs batteries de disques de tous types. L'une des informations que vous devez calculer est le nombre total de disques sélectionnés par l'utilisateur (voir ci-dessous).

model node	properties
 DiskArray SubModel	total_disks=sum(disks)
 100GB disk arrays	
 blue	disks=4
 mauve	disks=6
 250GB disk arrays	
 blue	disks=8
 mauve	disks=12

Supposons maintenant que vous vous rendez compte que vous avez également besoin de connaître le nombre de batteries de disques de 100 Go et le nombre de

batteries de disques de 250 Go. Au lieu de calculer ces valeurs en spécifiant les chemins d'accès vers les propriétés en question, ou en créant des règles qui doivent être associées à un point spécifique dans le modèle, ou en modifiant toutes les propriétés disk et total_disk, vous pouvez simplement définir quelques nouvelles propriétés qui ont les mêmes valeurs que l'ancienne propriété disk (voir ci-dessous).

model node	properties
DiskArray SubModel	total_disks=sum(disks) total_100GBdisks = sum(100GB_disks) total_250GBdisks = sum(250GB_disks)
<div style="margin-left: 20px;"> 100GB disk arrays <ul style="list-style-type: none"> blue disks=4 100GB_disks=4 mauve disks=6 100GB_disks=6 </div>	
<div style="margin-left: 20px;"> 250GB disk arrays <ul style="list-style-type: none"> blue disks=8 250GB_disks=8 mauve disks=12 250GB_disks=12 </div>	

Maintenant, si vous souhaitez connaître le nombre total de disques, vous pouvez toujours calculer cette valeur à l'aide de sum(disks). Vous pouvez également calculer les valeurs individuelles, et ce sans spécifier les chemins d'accès des propriétés individuelles ou modifier le travail que vous avez déjà fait.

Utilisation des feuilles de travail pour l'affectation des propriétés

Lors du développement d'un modèle, il est souvent nécessaire d'affecter le même ensemble de propriétés à plusieurs classes d'option ou éléments d'option. Les feuilles de travail sont très utiles dans ce cas, puisqu'elles permettent de définir rapidement les valeurs d'une propriété sur un nombre indéfini de classes d'option ou d'éléments d'option. Cela est particulièrement vrai lorsque vous utilisez une formule pour définir la valeur d'une propriété à plusieurs endroits. Le modélisateur peut simplement copier et coller la formule dans tous les éléments souhaités. Vous trouverez un exemple ci-dessous. Supposons que les propriétés d'affichage suivantes sont définies pour chaque élément dans un affichage tabulaire. Nous utilisons une feuille de calcul afin de pouvoir copier et coller facilement des formules figurant depuis les colonnes col1 et col2 vers chaque éléments de la classe d'option.

Nom de la feuille de travail		modelDisplay	
Ajouter ligne		Ajouter colonne	
Article	col1	col2	
30GB	$\$(expand("min_array_disk"))/\$(exapnd("max_array_"))$	$\$(expand("min_array_disk"))/\$(exapnd("max_array_"))$	
40GB	$\$(expand("min_array_disk"))/\$(exapnd("max_array_"))$	$\$(expand("min_array_disk"))/\$(exapnd("max_array_"))$	
75GB	$\$(expand("min_array_disk"))/\$(exapnd("max_array_"))$	$\$(expand("min_array_disk"))/\$(exapnd("max_array_"))$	

Un autre avantage des feuilles de travail réside dans la possibilité d'avoir un aperçu concis d'une section du modèle. Avec quelques notions de conception et de

planification, une feuille de travail peut fournir un aperçu d'une section du modèle ou une représentation complète de la solution à un problème spécifique. Vous trouverez ci-dessous un affichage différent de la même classe d'option, qui met en évidence les propriétés min et max définies pour chacun des éléments d'option.

modelMinMax				
Article	min array disk	max array disk	min cache memory	max cache memory
30GB	4	252	4	64
40GB	64	508	6	=value(exp_cache)
75GB	128	1148	6	128

Éviter le chaînage dans les formules de propriétés

Les propriétés associées à un élément ne comportent aucune notion de séquence. Cela signifie que lorsque vous utilisez des formules pour définir les valeurs des propriétés, vous ne pouvez pas vous appuyer sur un ordre particulier d'évaluation des formules. Si la propriété A contient une formule et que la propriété B contient une formule basée sur la propriété A, vous n'avez alors aucune garantie que la règle créée à partir de la formule B se déclenche après la règle créée pour la formule A. Pour contourner ce problème, le modélisateur a deux options :

- Transformer la première formule en règle qui se déclenche avant que la deuxième formule ne soit évaluée. Toutes les règles générées à partir des formules ont une priorité de 50. En créant une règle pour la première formule et en définissant sa priorité comme étant inférieure à 50, vous vous assurez que la valeur de la propriété A sera définie avant que la valeur de la propriété B ne soit calculée.
- Activer le déclenchement répété de règles. Dans ce cas, la première passe du déclenchement de la règle calculera la valeur de la propriété A. La seconde passe de la boucle de déclenchement de la règle calculera la valeur de la propriété B en fonction de la valeur de la propriété A calculée au cours de la première passe. Remarque : L'utilisation massive de ce type de chaînage de formules peut entraîner une dégradation des performances en raison du nombre de passes dans la boucle de déclenchement de la règle qui seront nécessaires pour satisfaire à toutes les conditions. Pour cette raison, nous recommandons la première option et conseillons de limiter le chaînage de formules autant que possible.

Définition des règles

Définition des conditions de règle

Les conditions de règle sont créées en appliquant des opérations booléennes aux expressions relationnelles. Une expression relationnelle est la comparaison entre une paire fonction/propriété et une autre paire fonction/propriété à l'aide d'opérateurs relationnels (inférieur à, égal à, supérieur à, dans, hors, etc.). Le résultat est vrai ou faux. Les opérateurs booléens tels que AND et OR encapsulent des ensembles d'expressions relationnelles. Les expressions relationnelles sont appelées *fragments*, étant donné que ce sont des fragments d'une règle. Le côté gauche de l'opérateur relationnel est souvent abrégé en LHS, et RHS représente le côté droit.

Classement des fragments de règle

L'évaluation des fragments de règle détermine le moment où une règle se déclenche ; l'ordre dans lequel les fragments apparaissent est donc essentiel. Plus le modèle peut déterminer rapidement si une règle est vraie ou fausse, plus il est efficace. Et bien évidemment, plus le modèle détermine rapidement qu'une règle ne doit pas être déclenchée, plus il peut continuer de traiter rapidement d'autres informations. Vous pouvez améliorer les performances en classant les fragments de règle du plus susceptible d'empêcher le déclenchement de la règle au moins susceptible d'empêcher le déclenchement de la règle.

Vous devez toujours tester vos règles pour vous assurer qu'elles ne se déclenchent que lorsque cela est nécessaire. Il est également important de connaître dans quelles circonstances les résultats d'une règle seront ou non utilisés. Par exemple, il est inefficace d'utiliser une règle d'extension qui se déclenche en permanence, mais qui n'effectue pas de sélection dans la section d'extensions si la formule de quantité est égale à zéro ou si les zones > et <= ne contiennent pas de correspondance pour la formule.

Création de règles générales

Lorsque cela est possible, créez des règles qui sont aussi générales que possible. Par exemple, la règle suivante peut être associée à n'importe quel élément auquel les propriétés `productType` et `handsetType` sont associées :

```
If propval(productType) != value(selectProductType)
and propval(handsetType) != value(phonePreference)
  set _isVisible=0
```

Cette règle se déclenche seulement pour les éléments auxquels la propriété `productType` est associée ET ne correspond pas aux types de produits sélectionnés, ET si les préférences de téléphone sélectionnées ne correspondent pas aux préférences de l'élément actuel. Une règle générale comme celle-ci peut remplacer des dizaines d'autres règles spécifiques comme les règles spécifiques suivantes :

```
If propval(productType) == literal("handset")
and propval(handsetType) != literal("camera")
and value(phonePreference) == literal("camera")
  set _isVisible=0
If propval(productType) == literal("handset")
and propval(handsetType) != literal("flip")
and value(phonePreference) == literal("flip")
  set _isVisible=0
...
```

Utilisation des formules pour définir des règles

Dans de nombreux cas, vous pouvez utiliser des formules à la place des règles. Lors de la modélisation, les formules sont conservées en tant que propriétés associées, qui ont comme valeur une expression évaluée au moment de l'exécution. Si l'une des fonctions référencées dans l'expression ne peut pas être évaluée, la formule agit comme une règle qui n'a pas été déclenchée. Si le déclenchement de la règle multipasse est activé, la formule est réévaluée à chaque passe de déclenchement jusqu'à ce que le déclenchement de la règle soit terminé ou que la formule produise son résultat.

Utilisez une formule plutôt qu'une règle lorsque la seule condition exigeant le calcul de la règle est que des valeurs soient associées à la fonction ou aux propriétés utilisées dans la formule.

Par exemple, supposons que vous souhaitez calculer le rayon de braquage pour les composants d'un camion comme l'essieu et l'empattement afin de garantir que le choix de l'utilisateur pour ces composants est judicieux. Vous pouvez définir une formule pour les composants du camion concernés afin de calculer le rayon de braquage `turningRadius` comme suit :

```
turningRadius = value(axelTurnFactor) * value(wheelBaseTurnFactor)*
sum(turningElements)
```

Cette formule se déclenchera lorsque chacune des expressions `value(axelTurnFactor)`, `value(wheelBaseTurnFactor)` et `sum(turningElements)` produiront toutes des résultats numériques.

La règle équivalente est la suivante :

```
if (value(axelTurnFactor) >= 0 or value(axelTurnFactor) < 0)
and (value(wheelBaseTurnFactor)>= 0 or value(wheelBaseTurnFactor)< 0)
and (sum(turningElements) >=0 or sum(turningElements) <0)
    turningRadius = value(axelTurnFactor) * value(wheelBaseTurnFactor)*
sum(turningElements)
```

La condition de la règle est plutôt longue et semble toujours renvoyer la valeur `True`. Toutefois, les fonctions peuvent renvoyer la valeur `NULL` si une propriété qu'elles référencent n'existe pas ; cette règle vérifie alors que le résultat n'est pas `NULL` en évaluant si la valeur renvoyée est `>= 0` ou `< 0`.

Éviter les informations de chemin dans les règles

Les expressions LHS et RHS d'un fragment de règle se composent d'un nom de fonction et d'un nom de propriété. Le nom de propriété peut contenir à la fois des informations sur le chemin d'accès absolu et relatif. Cependant, si vous spécifiez des informations sur le chemin d'accès d'une propriété dans un fragment de règle, la règle peut devenir inutilisable si les informations de chemin ou les classes d'option changent.

Par exemple, la règle suivante référence `wheelSize` et `wellSize` en utilisant des informations sur le chemin d'accès complet. Si le modélisateur a besoin de renommer les classes d'option correspondant aux roues ou au garde-boue ou s'il souhaite réutiliser la règle dans un autre modèle, la règle peut ne pas fonctionner correctement.

```
If value(*.wheels.wheelSize) == literal("17in")
and value(*.fender.wellSize) < literal(17)
    set _isVisible=0
```

Utilisez des informations de chemin uniquement pour accéder à une instance spécifique d'une propriété et, dans ce cas, uniquement s'il est impossible de créer un nouveau type de propriété pour stocker cette valeur. Si vous devez référencer le nom du chemin d'accès d'une propriété, il est souvent préférable d'utiliser des noms de chemins d'accès relatifs plutôt que des noms absolus.

Comparaison entre les tableaux de contraintes et les règles

La présente section explique les avantages et les inconvénients liés à l'utilisation des tableaux de contraintes pour limiter les choix des clients et à l'utilisation des règles. Les tableaux de contraintes limitent le choix d'un client sur une ou plusieurs options en se basant sur le choix du client sur une autre option. Par exemple, le choix d'une couleur extérieure pour une voiture peut limiter le choix des couleurs intérieures.

Les tableaux de contraintes fonctionnent mieux pour la validation simple, par exemple, pour déterminer si une option fonctionne ou ne fonctionne pas avec une autre option. Les tableaux de contraintes simples sont plus simples à gérer que les règles. Toutefois, des tableaux de contraintes volumineux et complexes peuvent être difficiles à gérer et entraîner des soucis en termes de performances.

Les tableaux de contraintes sont transformés en règles en interne.

Les règles conviennent mieux pour exprimer des problèmes de validation complexes et sont plus polyvalentes que les tableaux de contraintes. Les tableaux de contraintes et les règles peuvent afficher tous les deux des messages d'erreur, mais vous pouvez également créer des règles pour définir les propriétés ou faire des choix.

Amélioration des performances des modèles

La présente section explique certaines des techniques permettant de simplifier la création et la maintenance des modèles. L'utilisation de la technique appropriée peut avoir un impact important sur les performances du modèle.

- Utilisation de groupes de classes d'option, de groupes d'éléments d'option et de sous-assemblages :

Cette technique fonctionne bien lorsqu'un groupe d'options est répété dans plusieurs modèles différents.

Par exemple, supposez qu'une liste de disques durs à partir de laquelle l'utilisateur peut faire son choix est disponible pour chaque ordinateur que vous vendez. En créant des groupes de classes d'option, des groupes d'éléments d'option et des sous-assemblages, le modélisateur peut créer et maintenir des informations communes en un seul endroit, puis les utiliser dans de nombreux endroits.

Un des inconvénients de cette technique est qu'elle peut conduire à des modèles excessivement grands si un sous-assemblage est inclus dans le même modèle à plusieurs reprises.

- Technique de punch-in/punch-out de sous-modèle :

Cette technique est utile lorsqu'une configuration contient une sélection qui est également configurable. Vous pouvez utiliser la technique de punch-in/punch-out de sous-modèle pour imbriquer des configurations complexes dans un modèle de vente global.

L'inconvénient est que toutes les copies de l'élément configuré auront la même configuration.

- Instanciation dynamique :

Cette technique permet d'avoir plusieurs instances d'un élément configuré dans un seul modèle. Chaque instance peut avoir une configuration différente.

Outils de modélisation

La modélisation peut être un exercice long et fastidieux, mais au final, l'exactitude de la modélisation et l'évolutivité de la solution créée sont essentielles à la réussite du projet. Pour faciliter la création de modèles évolutifs et valides, nous avons développé une collection d'outils que les modélisateurs peuvent utiliser comme guide dans les diverses phases du développement. Au cours de l'étape de développement, les modélisateurs peuvent utiliser le journal de trace et l'outil de génération de rapports sur les modèles pour déterminer quels modèles doivent être débogués. Avant de mettre les modèles en production, leur évolutivité et leur stabilité peuvent être testées à l'aide de la plateforme de tests de charge. Pour

terminer, pendant l'exécution, la page d'état du cache du modèle fournit un aperçu de ressources système utilisées par le modèle, et l'analyseur de journal peut être utilisé pour interpréter la signification des informations du journal.

Utilisation du journal de trace

Le journal de trace contient les détails de l'exécution du moteur de règles. Il s'agit en général de la partie la plus fastidieuse de chaque demande adressée au serveur par le configurateur. Le journal de trace est conçu pour fournir les informations nécessaires pour le débogage des règles ayant un comportement anormal et contrôler le temps d'exécution des règles ; par conséquent, commencez toujours le débogage en examinant le journal de trace.

Vous pouvez créer un journal de trace à l'aide de Visual Modeler. Pour ce faire, procédez comme suit :

1. Accédez à Navigation dans le groupe de modèles, puis au modèle à déboguer.
2. Sélectionnez le modèle à partir du panneau Modèles et groupes.
3. Le modèle s'affiche dans l'onglet Aperçu du modèle.
4. Cliquez sur l'icône Test.
5. Le modèle s'exécute dans une fenêtre séparée.
6. Cliquez sur Déboguer.

Le journal de trace apparaît dans une fenêtre séparée.

Le journal est composé de deux sections. La première section est la trace de déclenchement de la règle et la deuxième section est le pool de propriétés tel qu'il est affiché la fin du déclenchement de la règle.

La figure suivante présente un exemple de section de trace du déclenchement de la règle.

Suivi de déclenchement de règle

#	(ms)	Résultat
0	0	Applying picks
1	0	Firing phase [0]:begin
2	0	Firing rules on MXDS-7500.Disk Drives
3	0	MSG_E_Available_HDD_Slots ==> fires on TRUE - priority = 50
4	0	Property not found [MX75_HDD_Ordered or MX75_Bays_Available], taking null action
5	0	took 0ms.
6	0	Firing rules on MXDS-7500.Memory
7	0	MSG_E_MX75_Memory_Software_Check ==> fires on TRUE - priority = 50
8	0	TESTING:sum(MX75_Mem_Ordered) <sum(MX75_Mem_Required) [nullreturn=false]
9	0	FALSE: 0.0 < 0.0
10	0	FALSE: sum(MX75_Mem_Ordered) <sum(MX75_Mem_Required) [nullreturn=false]
11	0	took 0ms.
12	0	Firing rules on MXDS-7500.Placeholder for auto memory selection
13	0	ASG make placeholder invisible ==> fires on TRUE - priority = 50
14	0	Left side property [MX75_Mem_Auto_Select] not found, taking null action
15	0	took 0ms.
16	0	Firing rules on MXDS-7500.AutoMemory
17	0	EXP_MX75_Automatic_Memory_Selection ==> fires on TRUE - priority = 50
18	0	Left side property [MX75_Mem_Auto_Select] not found, taking null action
19	0	took 0ms.
20	0	Firing rules on MXDS-7500.Software.Application
21	15	EXP_MX75_Fire_Wire ==> fires on TRUE - priority = 50
22	15	Left side property [MX75_Video_Editing] not found, taking null action
23	15	took 15ms.

La trace de déclenchement de la règle comporte trois colonnes :

- Un numéro de séquence, utile pour identifier la règle lorsque décrivez le problème rencontré à un tiers. Cela permet aux autres personnes de repérer rapidement le problème : "voir Xxx à la ligne 42".
- Le temps écoulé. Cela indique le temps écoulé entre le moment où l'entrée de journal a été créée et le moment où le déclenchement de la règle a démarré.
- Le corps du journal de trace. Ce dernier contient les détails du déclenchement de la règle, tels que la condition évaluée, les affectations qui peuvent se produire, le démarrage ou la fin d'un règle, etc.

Le journal présente le nombre de millisecondes nécessaires pour déclencher une règle après chaque entrée de déclenchement de règle. Le nombre total de millisecondes nécessaires pour exécuter le modèle est affiché à la fin de la trace de déclenchement de la règle.

La trace du pool de propriétés comporte également trois colonnes :

- Nom : Le chemin d'accès complet vers l'élément et la propriété de ce dernier.
- Type : Le type de propriété de la propriété nommée, comme par exemple Numérique, Liste ou Chaîne.
- Valeur : La valeur de la propriété une fois que la règle a été déclenchée.

La figure suivante présente un fragment d'un exemple de trace de pool de propriétés.

Pool de propriétés		
Nom	Type	Valeur
MXDS-7500.CONFIG: FIRST FIRE	Numérique	1.0
MXDS-7500.MX75_Bays_Available	Numérique	2.0
MXDS-7500.MX75_Card_Slot_Available	Numérique	4.0
MXDS-7500.MX75_Mem_Ordered	Numérique	0.0
MXDS-7500.MX75_Mem_Required	Numérique	0.0
MXDS-7500.Service Options.View Service.UI: COLUMN SPAN	Numérique	2.0
MXDS-7500.Placeholder for auto memory selection.UI: CONFIG CELL HTML CLASS	Chaîne	configsubcell_plain
MXDS-7500.Accessory Cards Message.Accessory Card Image.UI: CONFIG CELL HTML CLASS	Chaîne	configsubcell_plain
MXDS-7500.Microprocessor.UI: CONSTANT GUIDING TEXT	Chaîne	Dual processor capable motherboard, supporting Intel processors
MXDS-7500.Microprocessor.UI: CONTROL	Chaîne	RADIO
MXDS-7500.Disk Drives.UI: CONTROL	Chaîne	RADIO
MXDS-7500.Placeholder for auto memory selection.UI: CONTROL	Chaîne	controls/allpicked.jsp
MXDS-7500.Software.UI: CONTROL	Chaîne	CHECKBOX
MXDS-7500.Accessory Cards Message.UI: CONTROL	Chaîne	controls/allpicked.jsp
MXDS-7500.Accessory Cards Message.Accessory Card Image.UI: CONTROL	Chaîne	controls/allpicked.jsp
MXDS-7500.Accessory.Graphic Cards.UI: CONTROL	Chaîne	RADIO
MXDS-7500.Accessory.Cards.UI: CONTROL	Chaîne	CHECKBOX
MXDS-7500.Accessory.Network Cards.UI: CONTROL	Chaîne	RADIO
MXDS-7500.Service Options.View Service.UI: CONTROL	Chaîne	CHECKBOX
MXDS-7500.Placeholder for auto memory selection.UI: DEFAULT SELECTION	Chaîne	no
MXDS-7500.Accessory Cards Message.UI: DEFAULT SELECTION	Chaîne	no
MXDS-7500.Accessory Cards Message.Accessory Card Image.UI: DEFAULT SELECTION	Chaîne	no

Utilisez ce journal en mode utilisateur unique pour déterminer l'évolutivité des règles en fonction du temps de réponse par clic. Vérifiez combien il faut de temps pour déclencher les règles. Si la réponse est supérieure à 100-200 ms, vous pouvez rencontrer des problèmes d'évolutivité. Si c'est le cas, consultez le journal de trace pour vérifier si une règle particulière ne s'exécute pas correctement.

Utilisation de l'outil de génération de rapports sur les modèles

L'outil de génération de rapports sur les modèles fournit un aperçu de la taille d'un modèle par comparaison avec les autres modèles. Utilisez-le pour choisir les modèles à tester. Vous pouvez contrôler les résultats des tests au fil du temps afin de déterminer l'étendue des modifications apportées au modèle.

Utilisation des outils de test de charge

Les outils de test de charge permettent de déterminer le comportement de votre modèle une fois déployé. Avant d'utiliser les outils de test de charge, vous devez :

- Comprendre le comportement que vous souhaitez tester.
- Isoler vos scénarios de test pour estimer l'impact dans différentes configurations (test dans un réseau local et dans un réseau local distant, test avec et sans groupement, avec et sans façade Web, etc.).
- Être conscient que lorsque les modèles changent, il doit en être de même pour les scripts utilisés pour effectuer des tests et réexécuter des scénarios de test.

- Si les propriétés sont plus globales que le groupe de modèles en cours, définissez-les au niveau le plus bas de l'arborescence de groupes de modèles, qui est un ancêtre du modèle dans lequel vous souhaitez utiliser la propriété.

Suivi de déclenchement de règle

#	(ms)	Résultat
0	0	Applying picks
1	0	Firing phase [0]:begin
2	0	Firing rules on MXDS-7500.Disk Drives
3	0	MSG_E_Available_HDD_Slots ==> fires on TRUE - priority = 50
4	0	Property not found [MX75_HDD_Ordered or MX75_Bays_Available], taking null action
5	0	took 0ms.
6	0	Firing rules on MXDS-7500.Memory
7	0	MSG_E_MX75_Memory_Software_Check ==> fires on TRUE - priority = 50
8	0	TESTING:sum(MX75_Mem_Ordered) <sum(MX75_Mem_Required) [nullreturn=false]
9	0	FALSE: 0.0 < 0.0
10	0	FALSE: sum(MX75_Mem_Ordered) <sum(MX75_Mem_Required) [nullreturn=false]
11	0	took 0ms.
12	0	Firing rules on MXDS-7500.Placeholder for auto memory selection
13	0	ASG make placeholder invisible ==> fires on TRUE - priority = 50
14	0	Left side property [MX75_Mem_Auto_Select] not found, taking null action
15	0	took 0ms.
16	0	Firing rules on MXDS-7500.AutoMemory
17	0	EXP_MX75_Automatic_Memory_Selection ==> fires on TRUE - priority = 50
18	0	Left side property [MX75_Mem_Auto_Select] not found, taking null action
19	0	took 0ms.
20	0	Firing rules on MXDS-7500.Software.Application
21	15	EXP_MX75_Fire_Wire ==> fires on TRUE - priority = 50
22	15	Left side property [MX75_Video_Editing] not found, taking null action
23	15	took 15ms.

La trace de déclenchement de la règle comporte trois colonnes :

- Un numéro de séquence, utile pour identifier la règle lorsque décrivez le problème rencontré à un tiers. Cela permet aux autres personnes de repérer rapidement le problème : "voir Xxx à la ligne 42".
- Le temps écoulé. Cela indique le temps écoulé entre le moment où l'entrée de journal a été créée et le moment où le déclenchement de la règle a démarré.
- Le corps du journal de trace. Ce dernier contient les détails du déclenchement de la règle, tels que la condition évaluée, les affectations qui peuvent se produire, le démarrage ou la fin d'un règle, etc.

Le journal présente le nombre de millisecondes nécessaires pour déclencher une règle après chaque entrée de déclenchement de règle. Le nombre total de millisecondes nécessaires pour exécuter le modèle est affiché à la fin de la trace de déclenchement de la règle.

La trace du pool de propriétés comporte également trois colonnes :

- Nom : Le chemin d'accès complet vers l'élément et la propriété de ce dernier.
- Type : Le type de propriété de la propriété nommée, comme par exemple Numérique, Liste ou Chaîne.
- Valeur : La valeur de la propriété une fois que la règle a été déclenchée.

La figure suivante présente un fragment d'un exemple de trace de pool de propriétés :

Pool de propriétés		
Nom	Type	Valeur
MXDS-7500.CONFIG: FIRST FIRE	Numérique	1.0
MXDS-7500.MX75_Bays_Available	Numérique	2.0
MXDS-7500.MX75_Card_Slot_Available	Numérique	4.0
MXDS-7500.MX75_Mem_Ordered	Numérique	0.0
MXDS-7500.MX75_Mem_Required	Numérique	0.0
MXDS-7500.Service Options.View Service.UI: COLUMN SPAN	Numérique	2.0
MXDS-7500.Placeholder for auto memory selection.UI: CONFIG CELL HTML CLASS	Chaîne	configsubcell_plain
MXDS-7500.Accessory Cards Message.Accessory Card Image.UI: CONFIG CELL HTML CLASS	Chaîne	configsubcell_plain
MXDS-7500.Microprocessor.UI: CONSTANT GUIDING TEXT	Chaîne	Dual processor capable motherboard, supporting Intel processors
MXDS-7500.Microprocessor.UI: CONTROL	Chaîne	RADIO
MXDS-7500.Disk Drives.UI: CONTROL	Chaîne	RADIO
MXDS-7500.Placeholder for auto memory selection.UI: CONTROL	Chaîne	controls/allpicked.jsp
MXDS-7500.Software.UI: CONTROL	Chaîne	CHECKBOX
MXDS-7500.Accessory Cards Message.UI: CONTROL	Chaîne	controls/allpicked.jsp
MXDS-7500.Accessory Cards Message.Accessory Card Image.UI: CONTROL	Chaîne	controls/allpicked.jsp
MXDS-7500.Accessory.Graphic Cards.UI: CONTROL	Chaîne	RADIO
MXDS-7500.Accessory.Cards.UI: CONTROL	Chaîne	CHECKBOX
MXDS-7500.Accessory.Network Cards.UI: CONTROL	Chaîne	RADIO
MXDS-7500.Service Options.View Service.UI: CONTROL	Chaîne	CHECKBOX
MXDS-7500.Placeholder for auto memory selection.UI: DEFAULT SELECTION	Chaîne	no
MXDS-7500.Accessory Cards Message.UI: DEFAULT SELECTION	Chaîne	no
MXDS-7500.Accessory Cards Message.Accessory Card Image.UI: DEFAULT SELECTION	Chaîne	no

Utilisez ce journal en mode utilisateur unique pour déterminer l'évolutivité des règles en fonction du temps de réponse par clic. Vérifiez combien il faut de temps pour déclencher les règles. Si la réponse est supérieure à 100-200 ms, vous pouvez rencontrer des problèmes d'évolutivité. Si c'est le cas, consultez le journal de trace pour vérifier si une règle particulière ne s'exécute pas correctement.

Outil d'affichage de l'état du cache

Vous pouvez afficher les contenus actuels du cache en utilisant `cmd=configstatus`.

Performances

Règles de performances d'un modèle

- Chemins d'accès des éléments excessivement longs.
- Une règle qui ajoute de la mémoire en effectuant le calcul suivant :

$$\text{totalMem} = \text{value}(*.adapter.1.memory) + \text{value}(*.adapter.2.memory) + \text{value}(*.adapter.3.memory) + \text{value}(*.adapter.4.memory)$$
- sera beaucoup plus lente que :

$$\text{totalMem} = \text{sum}(memory)$$
- Si la propriété `memory` existe à d'autres endroits et a d'autres utilisations, ce qui est susceptible de générer une valeur non valide de `sum(memory)`, définissez

alors des propriétés supplémentaires sur les éléments d'adaptateur 1-4 appelés adapterMemory et utilisez la règle suivante :

`totalMem = sum(adapterMemory)`.

- Cela réclame beaucoup moins d'effort en termes de maintenance que :

`totalMem = value(*.adapter.1.memory) + value(*.adapter.2.memory) + value(*.adapter.3.memory) + value(*.adapter.4.memory)`

- Créez des règles qui se déclenchent uniquement lorsqu'elles sont nécessaires :
La règle `totalMem = sum(mem)` ne doit se déclencher que si `count(mem) > 0`.

Propriétés de performances des modèles

- Définissez les propriétés au niveau approprié dans la hiérarchie de groupes de modèles :
 - Si les propriétés sont locales uniquement sur ce modèle, définissez-les au niveau du modèle.
 - Si elles peuvent être utilisées par d'autres modèles au sein de ce groupe de modèles, définissez-les au niveau du groupe de modèles en cours.
 - Si elles sont plus globales que le groupe de modèles en cours, définissez-les au niveau le plus bas de l'arborescence de groupes de modèles, qui est un ancêtre du modèle dans lequel vous souhaitez utiliser la propriété.

Chapitre 4. Archivage des données

La gestion de vos données est primordiale pour protéger votre entreprise. Archivez régulièrement vos données de production et configurez des emplacements primaires et secondaires pour stocker les archives de vos bases de données, de préférence hors site. Établissez une planification régulière des activités d'archivage, incluant des sauvegardes incrémentielles quotidiennes et des sauvegardes complètes hebdomadaires (ou plus souvent si les volumes de votre entreprise l'exigent).

Chapitre 5. Mise à jour des statistiques

La mise à jour des statistiques de la base de données permet à l'optimiseur de requêtes de base de données de réexaminer les index de la base de données et recalculer les chemins les plus efficaces pour extraire les données. Cette section présente un script permettant de mettre à jour les statistiques pour une base de données Oracle et un autre permettant de mettre à jour les statistiques pour une base de données SQL Server.

Consultez votre administrateur de base de données pour mettre à jour correctement les statistiques des tableaux ou rapportez-vous à la documentation de votre base de données pour obtenir une aide supplémentaire.

Mise à jour des statistiques pour une base de données Oracle

L'exemple ci-après montre comment mettre à jour les statistiques d'une base de données Oracle au niveau du schéma. Remplacez *nom_schéma*, *nom_propriétaire* et *nom_tableau* par les noms de schéma, de propriétaire et de tableau appropriés.

```
EXEC DBMS_STATS.GATHER_SCHEMA_STATS(  
  ownname=> 'nom_schéma' ,  
  cascade=> TRUE,  
  estimate_percent=> DBMS_STATS.AUTO_SAMPLE_SIZE,  
  degree=> DBMS_STATS.AUTO_DEGREE,  
  granularity=>'AUTO',  
  method_opt=> 'FOR ALL COLUMNS SIZE AUTO');
```

L'exemple ci-après montre comment mettre à jour les statistiques d'une base de données Oracle au niveau du tableau.

```
exec dbms_stats.gather_table_stats(  
  ownname=> 'nom_propriétaire',  
  tabname=> 'nom_tableau',  
  estimate_percent=> DBMS_STATS.AUTO_SAMPLE_SIZE,  
  cascade=> DBMS_STATS.AUTO_CASCADE,  
  degree=> null,  
  no_invalidate=> DBMS_STATS.AUTO_INVALIDATE,  
  granularity=> 'AUTO',  
  method_opt=> 'FOR ALL COLUMNS SIZE AUTO');
```

Vous pouvez également mettre à jour les statistiques au niveau de la base de données ou des index, en fonction de vos exigences.

Mise à jour des statistiques pour une base de données SQL Server

L'exemple ci-après montre comment mettre à jour les statistiques d'une base de données c au niveau du tableau. Remplacez *nom_tableau* et *nom_index* par les noms de tableau et d'index appropriés.

```
UPDATE STATISTICS ON <nom_tableau> [ . <nom_index> ]  
  WITH FULLSCAN {, NORECOMPUTE }
```

Chapitre 6. Instructions concernant le réglage et la mémoire de la machine virtuelle Java

Lorsque vous rencontrez des problèmes liés à la mémoire, l'ajustement des paramètres de mémoire de la machine virtuelle Java permet de revenir à un environnement de travail sain. Cette section fournit des instructions de réglage des performances et des paramètres de mémoire de la machine virtuelle Java. Pour appliquer ces instructions, vous devez vous familiariser avec l'environnement de votre machine virtuelle Java et du conteneur de servlet.

Ajustement des paramètres de mémoire de la machine virtuelle Java

En règle générale, vous devrez attribuer autant de mémoire que possible à la machine virtuelle Java exécutant votre serveur d'applications. Pour ce faire, configurez la mémoire de la machine virtuelle comme suit :

- `-Xmx` devra occuper entre 80 % et 100 % de la mémoire physique de la machine. Si vous définissez une valeur `-Xmx` trop basse, le serveur d'applications peut être défaillant et afficher une erreur `OutOfMemory`. Si vous définissez une valeur `-Xmx` trop élevée, l'empreinte mémoire est plus grande et le segment de mémoire Java risque d'être remplacé, ce qui va générer d'autres problèmes en termes de performances.
- `-Xms` doit représenter environ la moitié du paramètre `-Xmx`. Si vous disposez de données historiques concernant le taux d'utilisation de la mémoire assurant la stabilité de votre serveur d'applications, vous pouvez alors définir `-Xms` autour de cette valeur.

Vous avez également la possibilité de définir `-Xms` sur le taux d'utilisation de la mémoire observé à la fin du servlet `InitServlet`. Cela garantira au minimum que l'initialisation DEBS s'exécutera avec une récupération de place aussi limitée que possible. Pour obtenir le taux d'utilisation de la mémoire, procédez comme suit :

1. Définissez `-Xms` sur la même valeur que `-Xmx`
2. Démarrez votre déploiement Visual Modeler et attendez que l'initialisation soit terminée.
3. Accédez à la page d'accueil de votre site d'e-commerce.
4. Ouvrez le fichier **debs.log** dans un éditeur de texte et recherchez une entrée du journal similaire à ce qui suit :

```
2003.03.18 ... END Request ... Mem=129380744/388726784/391291344 ...
```

5. Le premier nombre après **Mem=** correspond à l'utilisation actuelle de la mémoire après initialisation. Définissez `-Xms` sur ce nombre : dans l'exemple ci-dessus, utilisez la valeur `-Xms128m`.
- `-XX:MaxPermSize` contrôle la taille d'allocation pour les objets réflexifs similaires au système, tels que la classe et la méthode. Sa valeur initiale recommandée est 128m.

Pour les applications Web, l'espace attribué se remplit rapidement étant donné que les fichiers `*.jsp` sont convertis en fichiers `*.java`, puis en fichiers `*.class` qui sont chargés dans l'espace mémoire spécifié dans `-XX:MaxPermSize`. À partir de la version Java 1.4.2, vous pouvez utiliser `-XX:+PrintGCDetails` pour surveiller les détails de cet espace, appelé génération permanente.

Veillez à ne pas effectuer des modifications de la mémoire susceptibles de contredire les paramètres pris en charge actuellement par le serveur d'applications.

DEBS doit coexister avec le serveur d'applications dans la même machine virtuelle ; ainsi, en cas de doute, vérifiez attentivement la documentation du serveur d'applications ou contactez le service de support de votre serveur d'applications. Par exemple, supposez que la documentation du serveur d'applications utilisé stipule que le paramètre `-server` de la machine virtuelle Java n'est pas pris en charge. Dans ce cas, ne définissez pas le paramètre `-server`.

En dernier recours pour identifier et résoudre un problème, démarrez la machine virtuelle sans argument supplémentaire et ajoutez progressivement un argument, redémarrez, observez les résultats, puis ajoutez-en un autre, en poursuivant jusqu'à ce que vous ayez obtenu de bons résultats.

Réglage complémentaire des performances

Un réglage complémentaire des performances peut être effectué pour les activités de récupération de place Java et en ajustant les paramètres de mémoire pour d'autres zones telles que des unités d'exécution, des piles de la machine virtuelle Java ou des structures natives ou un code natif. Utilisez l'analyseur de journal ou observez directement le fichier `debs.log` pour déterminer les zones ayant des problèmes de performances.

Traçage des activités de récupération de place

Si vous remarquez des pauses inexplicables, il est possible que la machine virtuelle ait été interrompue pour une récupération complète de place. Pour confirmer cette hypothèse, utilisez le paramètre JVM `-verbose:gc` pour activer l'enregistrement des événements liés à la récupération de place dans le fichier `debs.log`. Les événements de récupération de place sont de plusieurs types :

```
[GC 325816K->83372K(776768K), 0.2454258 secs]
[Fu11 GC 267628K->83769K(776768K), 1.8479984 secs]
```

En général, une récupération mineure dure moins d'une demi-seconde. Une récupération majeure dure moins de trois secondes. Toute récupération de plus de trois secondes indique une condition hors plage et doit être examinée.

Les autres paramètres de traçage des activités de récupération de place que vous pouvez examiner sont les suivants :

- Le paramètre `-server` de la machine virtuelle Java permet d'adapter certains paramètres des segments de mémoire Java initiaux à un environnement de serveur. Définissez la valeur `-server` sauf si celle-ci n'est pas prise en charge par votre serveur d'applications.

Il existe un problème connu avec le paramètre `-server`, lié à un bogue se produisant dans la compilation JIT qui provoque la modification inattendue de la valeur utilisée par le service de données. De ce fait, DEBS ne pourra pas s'initialiser (échec du servlet `InitServlet`). Contactez votre représentant pour savoir comment désactiver la compilation JIT pour certaines méthodes.

L'utilisation du paramètre `-server` de la machine virtuelle est recommandée dans certains serveurs d'applications. Plus particulièrement, la valeur de `-XX:NewRatio` pour `-server` est 2 (la valeur par défaut du paramètre `-client` est 8). Pour plus d'informations sur les paramètres `-server` et `-client`, voir la documentation Sun à l'URL suivante :

http://java.sun.com/docs/books/performance/1st_edition/html/JAppHotspot.fm.html#998292

http://java.sun.com/docs/books/performance/1st_edition/html/JAppHotspot.fm.html#998359

- Le paramètre `-Xincgc` active la récupération de place incrémentielle. Le paramètre `-Xincgc` réduit les longs temps de pause occasionnés par la récupération complète de place. Lorsque vous utilisez ce paramètre, gardez à l'esprit que vous décalez le temps passé à réaliser une récupération majeure sur plusieurs récupérations mineures. Il existe des frais généraux associés à ce changement, qui représentent habituellement environ 10 %.
- Si vous recevez des messages d'erreur liés à l'insuffisance de mémoire, vous devez d'abord augmenter la valeur `-Xmx`, en vous assurant que cette valeur n'est pas supérieure à la valeur de la mémoire physique de la machine. Si vous recevez des messages d'erreur liés à l'insuffisance de mémoire lorsque le taux d'utilisation actuelle du segment de mémoire (si de nouveaux objets sont attribués) est loin de la valeur du paramètre `-Xmx`, d'autres zones d'allocation de mémoire sont peut-être saturées. Examinez le rapport de l'analyseur de journal et déterminez si les erreurs proviennent des zones suivantes :
 - En raison des classes : essayez de définir le paramètre `-XX:MaxPermSize=128m`
 - En raison des unités d'exécution : essayez d'ajuster la pile en définissant le paramètre `-Xss=512k`
 - En raison des piles de la JVM : essayez d'ajuster la pile en définissant le paramètre `-Xss=512k`
 - En raison des structures de données natives : essayez d'ajuster la taille de la zone d'échange du système d'exploitation
 - En raison des codes natifs : essayez d'ajuster la taille de la zone d'échange du système d'exploitation

Chapitre 7. Analyseur de journal

L'analyseur de journal est un outil open source permettant d'analyser les entrées du fichier **debs.log** de Visual Modeler. L'outil permet de visualiser les indicateurs clés de performances : les unités d'exécution, la mémoire, les demandes et les sessions ainsi que les temps de réponse triés par utilisateur et type de demande.

En utilisant l'analyseur de journal dans le cadre d'une routine quotidienne de surveillance de votre déploiement, vous bénéficiez des avantages suivants :

- Les rapports quotidiens de l'analyseur de journal permettent d'augmenter la fiabilité et la stabilité de votre déploiement. Les données générées peuvent vous avertir à l'avance d'un problème potentiel, ce qui permet d'éviter les indisponibilités. Par exemple, en utilisant les rapports quotidiens de l'analyseur de journal, vous pouvez prévoir de manière proactive de redémarrer un serveur d'applications lorsqu'il approche de sa pleine capacité de mémoire.
- Les rapports quotidiens de l'analyseur de journal servent de base pour l'identification et la résolution d'un problème en cours. En examinant les rapports, vous pouvez déterminer à quel moment le problème a commencé et le mettre en corrélation avec certains événements, telle qu'une mise à niveau du système d'exploitation.
- Les rapports quotidiens de l'analyseur de journal fournissent un point focal pour apporter des améliorations incrémentielles. En révisant le rapport de l'analyseur de journal chaque jour, vous pouvez générer une liste des tâches pour prévoir quand redémarrer votre serveur d'applications, nettoyer les listes d'exceptions, détecter les unités d'exécution bloquées ou fournir des commentaires en retour aux développeurs au sujet des demandes à exécution longue ou des demandes qui utilisent une quantité considérable de ressources, comme par exemple renvoyer des lignes de grande taille à partir d'une base de données.

Contactez votre représentant pour obtenir l'analyseur de journal. L'analyseur de journal est un fichier **.jar** qui peut être enregistré et décompressé à n'importe quel emplacement. L'analyseur de journal attend que les entrées du journal DEBS soit au format suivant :

```
<AAAA.MM.JJ HH:MM:SS:mss NomUnitéExécution:NiveauConsignation:BaliseJournal:messages>
```

Par exemple :

```
2006.10.12 06:00:00:171 Env/http-8580-Processor48:INFO:WrappingFilter ...
```

Étant donné que le processus d'analyse d'un fichier journal peut consommer beaucoup de mémoire, spécifiez autant de mémoire JVM que possible afin d'éviter les erreurs liées à une insuffisance de mémoire. Par exemple, démarrez l'analyseur de journal comme suit :

```
java -Xms256m -Xmx512m -jar logAnalyzer-1.1.1-SNAPSHOT-app.jar
```

L'écran initial de l'analyseur de journal s'affiche.

Entrez les informations suivantes :

- **Source (Source)** : Entrez le chemin d'accès complet d'un fichier journal DEBS ou d'un répertoire contenant plusieurs fichiers journaux DEBS.
- **Input (Entrée)** :

- **DEBS 6.4 or later (DEBS 6.4 ou version ultérieure)** : Vérification automatique. Si vous analysez un fichier journal d'une version antérieure à la version 6.4 de Visuel Modeler, décochez cette case.
- **From WebLogic (Depuis WebLogic)** : Cochez sur cette case pour indiquer que les fichiers journaux sont générés à partir d'un serveur d'applications Oracle WebLogic.
- **Output (Sortie)** : Cochez cette case pour générer un graphique des temps de réponse regroupés par type de message.
- **Output dir (Rép. sortie)** : Entrez le chemin d'accès complet nom de chemin absolu nom de chemin du répertoire qui doit contenir la sortie du rapport.

Cliquez sur **Start analyzing (Démarrer l'analyse)** pour lancer le processus d'analyse du journal. L'analyseur de journal affiche des messages au fur et à mesure de l'analyse puis, une fois l'analyse terminée, place la sortie de l'analyse du journal dans le répertoire de sortie spécifié.

Configuration des rapports quotidiens de l'analyseur de journal

La présente section décrit une procédure permettant d'automatiser la génération quotidienne des rapports de l'analyseur de journal. La procédure décrite ici utilise l'outil Ant, puisque ce dernier est portable, couramment utilisé et possède une bonne documentation. L'outil Ant est disponible sur <http://ant.apache.org>.

Les objectifs de cette procédure sont les suivants :

- Configurer un travail cron pour exécuter les rapports pendant la nuit et organiser une sortie par date (année/mois/jour) pour faciliter la navigation.
- Compresser les fichiers journaux dans la mesure du possible pour économiser de l'espace.
- Configurer l'automatisation de manière à faciliter sa duplication, afin que les fichiers journaux provenant de plusieurs déploiements puissent être hébergés à partir d'un serveur unique.

Pour automatiser la génération des rapports de l'analyseur de journal, vous avez besoin de :

- Java et Ant
- Un accès en lecture aux fichiers journaux DEBS
- Un accès en écriture au répertoire de sortie des rapports, *<out.dir>*. Les contenus du répertoire *<out.dir>* sont accessibles via un serveur Web.

Automatisation des rapports quotidiens de l'analyseur de journal

Pourquoi et quand exécuter cette tâche

La présente section décrit le flux de travail général pour l'automatisation des rapports quotidiens de l'analyseur de journal.

Procédure

1. DEBS génère des journaux dans le répertoire **logs** du serveur d'applications ou du conteneur de servlet.

Par exemple, dans un déploiement Tomcat, le répertoire **logs** se trouve dans *<tomcat-home>\logs*.

Le journal est appelé **debs.log.n**, où *n* correspond à un numéro. Par exemple, `debs.log.1`, `debs.log.2`, et ainsi de suite.

2. Configurez l'exécution quotidienne d'un travail cron (de préférence tôt le matin) pour concaténer tous les journaux du répertoire de journaux dans un fichier temporaire.
3. À partir du fichier temporaire, extrayez les entrées de journal de la veille vers le répertoire de sortie de l'analyseur de journal en utilisant le modèle de désignation de répertoire `année/mois/jour/log.suffixe`.
4. Le fichier **année/mois/jour/log.suffixe** est ensuite compressé à l'aide de l'outil **gzip** pour économiser de l'espace.
5. Démarrez l'analyseur de journal pour faire une analyse syntaxique du fichier **année/mois/jour/log.suffixe.gz** et générer le rapport vers le répertoire `année/mois/jour/html/`.

Configuration des rapports quotidiens de l'analyseur de journal

Procédure

1. Si vous ne l'avez pas déjà fait, contactez votre représentant afin d'obtenir les fichiers suivants :
 - le fichier **.jar** de l'analyseur de journal
 - **logAnalyzer-daily.xml**
 - `logAnalyzer-daily.properties`
2. Enregistrez les fichiers de l'analyseur de journal dans un emplacement temporaire.
3. Voir «Configuration de l'analyseur de journal», à la page 38 pour des informations sur les valeurs de configuration.
4. Utilisez la commande suivante pour exécuter le rapport quotidien de l'analyseur de journal :

```
ant -Dproperties.file.name="logAnalyzer-daily.properties" -f logAnalyzer-daily.xml
```
5. Examinez la sortie. L'emplacement de sortie est similaire à celui qui suit :

```
sites/default/app-server/logAnalyzer-out.d/dailySplit/AAAA/MM/JJ/html/index.html
```

Recommandations concernant la structure du répertoire

La figure suivante présente les recommandations concernant la structure du répertoire de l'analyseur de journal. Cette structure est particulièrement recommandée si vous prévoyez d'héberger des fichiers journaux provenant de plusieurs sites.

```

# where to keep the log analyzer jar file
bin/
  logAnalyzer-1.1.1-SNAPSHOT-app.jar

# ant script
logAnalyzer-daily.xml

# sites data
sites/
  site1/
  ...
  site2/
  ...
  siteN/
    logAnalyzer-daily.properties
  app-server/
    logs/
      debs.log
      debs.log.1
      debs.log.2

```

Les informations sur le site sont conservées dans le répertoire **sites**, qui contient un répertoire pour chaque site. Le nom du répertoire du site peut être une chaîne unique ; l'exemple ci-dessous utilise le nom *siteN*, où *n* est un numéro : *site1*, *site2*, etc.

Chaque répertoire du site contient un fichier **logAnalyzer-daily.properties** qui contient les paramètres spécifiques du site.

Les fichiers journaux de chaque site sont conservés dans le répertoire **siteN/app-server/logs/**.

Le répertoire **sites** est en lecture seule. La sortie est écrite dans le répertoire **siteN/app-server/logAnalyzer-out.d**.

En utilisant la structure ci-dessus, vous pouvez lancer un travail cron à l'aide du nom du site uniquement. Par exemple, pour un site qui s'appelle *bbfb-01* :

```

# Indiquer à Ant de définir la valeur de site.name et d'utiliser un nom de script de génération :
# logAnalyzer-daily.xml
ant -Dsite.name=bbfb-01 -f logAnalyzer-daily.xml

```

Si vous renommez **logAnalyzer-daily.xml** en **build.xml**, vous pouvez omettre l'argument **-f logAnalyzer-daily.xml**. Par exemple, pour un site qui s'appelle *bbfb-01* :

```

ant -Dsite.name=bbfb-01

```

Configuration de l'analyseur de journal

Les paramètres spécifiques au déploiement sont définis dans un fichier de propriétés. Le fichier de propriétés par défaut est **sites/\${nom.site}/logAnalyzer-daily.properties**. Vous pouvez également définir le nom du fichier de propriétés à partir de la ligne de commande comme suit :

```

ant -Dproperties.file.name="chemin_du_fichier.properties" ...

```

Vous trouverez ci-dessous une liste des propriétés de configuration du fichier **logAnalyzer-daily.properties**.

- **log.dir** : Le chemin d'accès complet vers l'emplacement du répertoire contenant les fichiers journaux DEBS. Par exemple :

```
# le répertoire par défaut est ./logs
log.dir=/home/hle/tomcat/logs
```
- **out.dir** : L'emplacement où vous souhaitez écrire les rapports générés. Par exemple :

```
# le répertoire par défaut est logAnalyzer-out.d
out.dir=/home/hle/public_html/logAnalyzer-out.d
```
- **logAnalyzer.jar** : L'emplacement du fichier **.jar** de l'analyseur de journal. Par exemple :

```
# le répertoire par défaut est ./logAnalyzer-1.1.1-SNAPSHOT-app.jar
logAnalyzer.jar=target/logAnalyzer-1.1.1-SNAPSHOT-app.jar
```
- **is.weblogic** : A la valeur True si les fichiers journaux ont été générés par WebLogic. Par exemple :

```
# la valeur par défaut est False
is.weblogic=true
```
- **genChart.perMessageType** : Spécifiez la valeur False pour ignorer la génération des graphiques messageType. Par exemple :

```
# la valeur par défaut est True
genChart.perMessageType=false
```
- **log.prefix** : Le préfixe du journal DEBS. Il est très rare que vous ayez à modifier cette valeur. Par exemple :

```
# la valeur par défaut est debs.log
log.prefix=Midwest.log
```
- **target.date.offset** : Définit automatiquement la valeur target.date. La valeur par défaut est 1, ce qui correspond à la date de la veille. Par exemple, définissez target.date.offset sur 7 pour extraire les fichiers journaux pour la semaine précédente :

```
# la valeur par défaut est 1 (date de la veille)
target.date.offset=7
```
- **target.date** : Limite le traitement aux entrées de journal pour ce jour. L'usage le plus courant pour ce paramètre est de régénérer manuellement un ancien ensemble de fichiers journaux. Par exemple :

```
# la valeur par défaut correspond à la date de la veille (évaluée automatiquement)
target.date=2006/07/24
```

Index

A

- allocation de mémoire
 - zones à vérifier 33
- analyse du fichier debs.log 35
- analyseur de journal 35
 - fichier de propriétés 38
 - structure de répertoire 37
- architecture de déploiement 1
 - environnement de construction 1
 - zone d'assurance qualité 1
 - zone de transfert 2
- archivage des données 27
- automatisation des rapports de l'analyseur de journal 36

D

- données
 - diagramme pour la restauration 3
 - protection 3

I

- indicateurs clés de performances 35
- infrastructure
 - niveau application 2
 - niveau base de données 2
 - niveau Web 2
 - typique 2

M

- machine virtuelle Java (JVM)
 - server 32
 - verbose
 - gc 32
 - Xincgc 32
- mise à jour des statistiques
 - Oracle 29
 - SQL Server 29
- modèle, taille 8

N

- noms des propriétés 12

P

- paramètre -Xmx 33
- performances
 - erreur liée à l'insuffisance de mémoire 33
 - problèmes de mémoire 31
- politiques de restauration 4
- propriétés 12
 - définition au niveau approprié 25

R

- récupération de place incrémentielle
 - Xincgc 33
- redondance 2
- réglage des performances
 - récupération de place 32

S

- sauvegarde et restauration, stratégies 3
- sauvegardes
 - complètes 3
 - sauvegarde de point de contrôle 3
 - sauvegarde incrémentielle 3
 - sauvegarde quotidienne 3
 - sauvegardes hebdomadaires 3
- scénario de récupération 4
- sous-modèles 9
- stratégies
 - sauvegarde et restauration 3

Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales : LE PRÉSENT DOCUMENT EST LIVRÉ "EN L'ÉTAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DÉCLINE NOTAMMENT TOUTE RESPONSABILITÉ RELATIVE À CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DÉFAUT D'APTITUDE À L'EXÉCUTION D'UN TRAVAIL DONNÉ. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les programmes exemples sont fournis "en l'état", sans garantie d'aucune sorte. IBM ne sera en aucun cas responsable des dommages liés à l'utilisation de ces programmes exemples.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© IBM 2011. Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. 2011.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Marques

IBM, le logo IBM et [ibm.com](http://www.ibm.com) sont des marques d'International Business Machines Corp. dans de nombreux pays. Les autres noms de produits et de services peuvent appartenir à IBM ou à des tiers. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web "Copyright and trademark information" à l'adresse <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, le logo Adobe, PostScript et le logo PostScript sont des marques d'Adobe Systems Incorporated aux États-Unis et/ou dans certains autres pays.

IT Infrastructure Library est une marque de The Central Computer and Telecommunications Agency qui fait désormais partie de The Office of Government Commerce.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, et Pentium sont des marques d'Intel Corporation ou de ses filiales aux États-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds aux États-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux États-Unis et/ou dans certains autres pays.

ITIL est une marque de The Office of Government Commerce et est enregistrée au bureau américain Patent and Trademark Office.

UNIX est une marque enregistrée de The Open Group aux États-Unis et/ou dans certains autres pays.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses filiales.

Cell Broadband Engine est une marque de Sony Computer Entertainment, Inc. aux États-Unis et/ou dans certains autres pays, et est utilisée sous license.

Linear Tape-Open, LTO, le logo LTO, Ultrium et le logo Ultrium Logo sont des marques de HP, IBM Corp. et Quantum aux États-Unis et/ou dans certains autres pays.

Connect Control Center, Connect:Direct, Connect:Enterprise, Gentran, Gentran:Basic, Gentran:Control, Gentran:Director, Gentran:Plus, Gentran:Realtime, Gentran:Server, Gentran:Viewpoint, Sterling Commerce, Sterling Information Broker et Sterling Integrator sont des marques de Sterling Commerce, Inc., une filiale d'IBM Company.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.



Imprimé en France